Project: ShelfSense – Intelligent In-Store Inventory Replenishment System

1. Introduction

This document outlines the Low-Level Design (LLD) for **ShelfSense**, an intelligent inventory replenishment system designed for large-format retail stores. The system automates shelf stock monitoring, predicts replenishment needs, and streamlines coordination between store staff and warehouse operations.

Supports backend development using **Java (Spring Boot)** and **.NET (ASP.NET Core)**.

2. Module Overview

3. Architecture Overview

3.1 Architectural Style

- **Frontend**: Angular/React for store dashboards and staff task views.
- **Backend**: REST APIs for inventory prediction and task orchestration.
- **Database**: PostgreSQL/MySQL for structured product and stock data.

3.2 Component Interaction

- Frontend displays shelf status and alerts.
- Backend processes stock levels and generates replenishment tasks.

4. Module-Wise Design

4.1 Product Catalog & Shelf Mapping Module

4.1.1 Features

- Map products to shelf locations and categories.
- Track shelf capacity and restocking frequency.

4.1.2 Entities

- **ProductShelf**
    - ProductID
    - ShelfID
    - Category
    - MaxCapacity
    - CurrentStock

4.2 Replenishment Prediction & Alerting Module

4.2.1 Features

- Predict stock depletion based on sales velocity.
- Generate alerts for low-stock items.

4.2.2 Entities

- **ReplenishmentAlert**
    - AlertID
    - ProductID
    - ShelfID
    - PredictedDepletionDate
    - UrgencyLevel

4.3 Store Staff Task Management Module

4.3.1 Features

- Assign restocking tasks to staff.
- Track task completion and delays.

4.3.2 Entities

- **RestockTask**
    - TaskID
    - ProductID
    - ShelfID
    - AssignedTo
    - Status

4.4 Warehouse Coordination Module

4.4.1 Features

- Request stock from warehouse based on shelf alerts.

- Track delivery status and expected arrival.

4.4.2 Entities

- **StockRequest**
  - RequestID
  - ProductID
  - Quantity
  - RequestDate
  - DeliveryStatus

4.5 Inventory Analytics & Stockout Reporting Module

4.5.1 Features

- Analyze restocking efficiency and shelf availability.
- Generate reports on stockouts and replenishment delays.

4.5.2 Entities

- **InventoryReport**
  - ReportID
  - ShelfID
  - ProductID
  - StockoutFrequency
  - GeneratedDate

5. Deployment Strategy

5.1 Local Deployment

- Developer machines with simulated shelf data.

5.2 Staging and Production Environments

- Cloud deployment with centralized inventory access.

6. Database Design

6.1 Tables and Relationships

- ProductShelf → ReplenishmentAlert → RestockTask → StockRequest → InventoryReport

7. User Interface Design

7.1 Wireframes

- Shelf Dashboard: Visual shelf status and alerts.
- Task Board: Staff assignments and progress.
- Analytics Console: Stockout trends and performance metrics.

8. Non-Functional Requirements

8.1 Performance

- Support 1,000 concurrent users across multiple stores.

8.2 Usability

- Mobile-friendly interface for store staff.
- Color-coded alerts and task filters.

8.3 Security

- Role-based access for store managers and staff.
- Secure API tokens for warehouse integration.

8.4 Scalability

- Support for multi-store chains and regional warehouses.

9. Assumptions and Constraints

9.1 Assumptions

- Stores maintain digital records of shelf stock.
- Warehouse systems are API-enabled.

9.2 Constraints

- Initial rollout for FMCG and grocery sections only.